Deployment Guidance

This document provides instructions on how to deploy and set up the FL-Koala backend and frontend environment.

FL-Koala Backend Setup Guide

This section includes instructions on how to set up the FL-Koala backend environment via Docker Compose.

> ⚠ **Prerequisites**
>
> - Python 3.12
> - Docker and Docker Compose (for Docker setup)
> - PostgreSQL >=15.6 (for database setup)

Setup Using Docker Compose

1. Build and Run with Docker Compose

Ensure you have Docker and Docker Compose installed. Open a terminal: Set the `DATABASE_URL` on your host machine:

- On Linux/Mac:

```
export DATABASE_URL="postgresql://yourusername:yourpassword@localhost:5432/yourdbname"
```

- On Windows (Command Prompt):

```
set DATABASE_URL="postgresql://yourusername:yourpassword@localhost:5432/yourdbname"
```

2. Ensure PostgreSQL is Running

Make sure PostgreSQL is installed and running on your server. If not installed, you can follow the installation guide for your specific operating system.

3. Inspect SQL Files

- `FL-Koala/src/backend/migrate/schema.sql`: Contains the SQL commands to create tables and schema.
- `FL-Koala/src/backend/migrate/seed.sql`: Contains the SQL commands to create sample scripts and users.

4. Run SQL Scripts

Before running the Docker containers, you need to initialize the database. You can do this by running the following commands:

- On Linux/Mac:

```
psql -U yourusername -d yourdbname -a -f schema.sql
psql -U yourusername -d yourdbname -a -f seed.sql
```

- On Windows (Command Prompt):

```
psql -U yourusername -d yourdbname -a -f schema.sql
psql -U yourusername -d yourdbname -a -f seed.sql
```

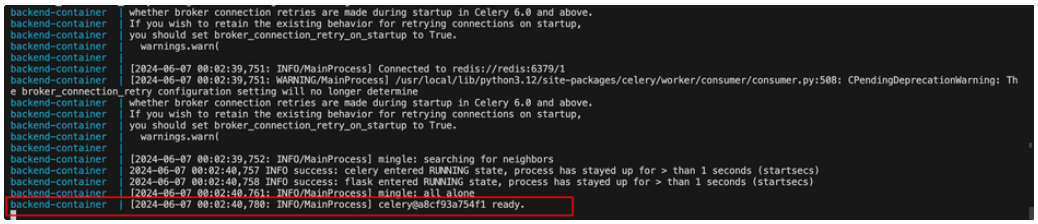
5. Run Docker Compose

Navigate to the project directory (`~/FL-Koala/src/backend`) and run:

```
docker-compose up --build
```

This command builds the Docker image and starts the service, exposing it on port 8080.

Docker is beginning to build, please wait

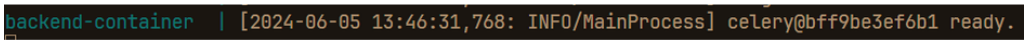


You will see the message when it runs successfully

Rebuilding the Image

If you need to rebuild the image, run:

```
docker-compose up --build
```



Suceessfully rebuilt

Summary

- Ensure PostgreSQL is running and accessible.
- Use `schema.sql` for schema setup and `seed.sql` for initial data.
- Properly set `DATABASE_URL`.

FL-Koala Frontend Setup Guide

Before setting up the frontend, ensure that your environment meets the following prerequisites:

> **Prerequisites**
>
> - Node.js >= 16.x
> - npm >= 8.x

You can download and install Node.js and npm here Node.js — Run JavaScript Everywhere

Package Managers

- It is recommended to use `yarn` or `pnpm` as the package manager
- Install yarn:

```
npm install --global yarn
```

- Install pnpm:

```
npm install --global pnpm
```

Setup Frontend

Navigate to (`~/FL-Koala/src/frontend`) and run commands:

1. Dependency Installation

Run the following command in the project root directory to install the required dependencies:

```
yarn install
```

Or using pnpm:

```
pnpm install
```

2. Run the server

```
yarn run dev
```

Or using pnpm:

```
pnpm run dev
```

Ready message poping-up means you successfully build frontend.

3. Open the browser

After “Ready” message pops up, navigate to `http://localhost:3000` in any local brower

Q&A:

Q: How can I install new python libraries that are used in scripts?

A: Update `~/FL-Koala\src\backend\requirements.txt` so that the backend service can run scripts properly.

Q: How can I clear up the input and output space?

A: Inspect `~/FL-Koala\src\backend\data` to remove unnecessary files.

Q: Why the building commond `docker-compose up --build` doesn’t work for me?

A: Please check if you navigate the right project directory `~/FL-Koala/src/backend`

Common issues

Node.js Version Issues

If you encounter version mismatch issues, you can use `nvm` to manage and switch between different Node.js versions:

```
nvm install 10
nvm use 10
```

Current developer's environment version: